Part VII

METATHEORY AND THE SCOPE AND LIMITS OF LOGIC

20

Metatheory

ALASDAIR URQUHART

1 Introduction

The older tradition in mathematical logic, represented by the foundational work of Frege, Whitehead, and Russell, was one in which the aim was to develop as large a part of mathematics as possible within a fixed axiomatic system. In general, questions that fell outside the basic system (such as the system of type theory on which *Principia Mathematica* is based), were ignored.

Under the influence of the great German mathematician David Hilbert, a new approach became influential in the 1920s, sometimes called 'metamathematics' or 'metalogic.' This new approach, in contrast with the earlier, can be described as critical in spirit, both in the sense that the underlying ideas showed a strong Kantian influence, but also in that the trend was towards analysing logical systems from the outside, rather than working within a fixed system of axioms. As a consequence of this change in direction, logic became a much more mathematical subject than formerly, a trend that continues to this day. The results that emerged from the research program of the Hilbert school remain among the most striking in all of logic.

In the present chapter, we describe these results in non-technical language, and indicate their philosophical significance. They are in many cases of a negative character, showing that the optimistic goals of Hilbert's foundational program could not be achieved. Nevertheless, a central concept emerged from this research activity, that of computability. The truly remarkable fact that this concept, in contrast to notions like that of provability and definability, does not depend on the system with respect to which it is defined, but is in a certain sense absolute, is fundamental to modern computer science and technology.

We begin with an outline of Hilbert's program in the foundation of mathematics, the achievements of Gödel that contributed positively to Hilbert's aims (the completeness theorem) and results like his incompletenesss theorem that showed the original aims to the program to be untenable, and led to its demise, at least in its original form. The essay also discusses the concept of computability that emerged in the 1930s in the wake of the incompleteness theorems, and the resulting clarification of the extent to which logic can be considered a purely formal subject. It concludes

with a discussion of the philosophical bearings of the basic results, in particular the question of the absolute or relative nature of logical concepts.

2 Hilbert's Program

David Hilbert (1862–1943) dominated German mathematics in the first half of the twentieth century. His formalist program, which attained its classical formulation in the 1920s, was intended to provide a final solution to the foundational problems that had arisen in the wake of the debates in the foundations of set theory, and the constructivist criticisms of the Dutch intuitionist L. E. J. Brouwer (1881–1966).

Brouwer had severely criticized the free use of classical logic as applied to infinite structures, and in particular the law of excluded middle. Given a constructive reading of the logical particles, the law of excluded middle can be read as asserting the universal solubility of all mathematical problems (that is to say, 'P or not P' asserts that we have either a constructive proof of the proposition P, or a constructive refutation of P). Since there is no warrant for this belief, Brouwer rejects the applicability of classical logic in general.

Hilbert presented himself as the champion of classical methods in mathematics, making such ringing assertions as the following from an address of 1927:

> Taking the principle of excluded middle from the mathematician would be the same, say, as proscribing the telescope to the astronomer or to the boxer the use of his fists.
>
> (van Heijenoort 1967: 476)

Hilbert spelled out his program in detail in a series of addresses from the 1920s that can be found in translation in van Heijenoort's collection of basic logical texts. He accepted (in a sense) the constructivist criticism of classical logic, since he denies the existence of the actual infinite. However, he wished to keep the powerful deductive tools of classical logic and set theory, and so was forced to adopt an indirect strategy of justification.

The essentials of Hilbert's formalist program are as follows. Classical mathematics is to be given a complete and fully rigorous formulation by employing the resources of mathematical logic (making use of the work already done in this area by Frege, Whitehead, and Russell). However, not all the statements occurring in such systems are held to be directly meaningful. In particular, purely existential statements are to be read as infinite disjunctions, and so we cannot directly attribute a constructive meaning to them. The part of mathematics that is directly meaningful for Hilbert he describes as the part consisting of finitary inferences, operating on concrete objects consisting of strings of symbols.

> If logical inference is to be reliable, it must be possible to survey these objects completely in all their parts, and the fact that they occur, that they differ from one another, and that they follow each other, or are concatenated, is immediately given intuitively, together with the objects, as something that neither can be reduced to anything else nor requires

> reduction. This is the basic philosophical position that I consider requisite for mathematics and, in general, for all scientific thinking, understanding and communication.
>
> (van Heijenoort 1967: 376)

Thus the main goal of Hilbert's program can be stated as the solution of the *consistency problem*. We can regard the symbol sequences constituting the formalized version of mathematical assertions as purely formal objects. It is then a mathematically well-defined problem to show that a sequence of such objects satisfying the (purely formal) conditions for being a correct proof cannot end with an obvious contradiction like '0 = 1.' To be a fully convincing demonstration, and avoid the charge of circularity, the proof must itself be based only on finitary reasoning. Hilbert hoped that by completing such a consistency proof he would achieve a final vindication of classical mathematics.

Hilbert was a congenital optimist, and in particular believed strongly in the solvability of all mathematical problems, a faith that expressed itself in the phrase he used as the conclusion of his last major public address: "We must know. We shall know" (Ewald 1996: 1165). This credo forms the background to another major problem of the Hilbert school, the *Entscheidungsproblem* or decision problem. The problem here is to decide by a mechanical, algorithmic procedure for a given formula of first-order predicate logic, whether it is logically valid or not.

If there were a positive solution to this problem, this would have extraordinarily far-reaching consequences. In particular, all known mathematical theories can be formalized in terms of finite sets of axioms in first-order logic. If the decision problem were solvable, then it would be possible for any such theory to decide whether a given sentence is a theorem simply by forming the implication that has the conjunction of the axioms as the antecedent, and the sentence as the consequent, and testing this implication for validity. Hence, all mathematical problems would be solvable in principle by a purely mechanical procedure. Thus Hilbert's belief in the solvability of all problems would be true, and what is more, in an extremely strong sense, since arbitrary problems could be solved without the intervention of human ingenuity.

A final problem that is of a somewhat subsidiary character, but fits naturally into Hilbert's formalist viewpoint, is the problem of completeness for first-order predicate logic. The problem was originally posed by Hilbert and Ackermann in their textbook of logic of 1928. We can define validity in predicate logic in two different ways, syntactically or semantically. The first definition of validity defines it in terms of derivability from a fixed set of axioms or rules, such as those originally proposed by Frege. The second definition defines it as truth in all possible interpretations. The second notion is not a finitistically meaningful notion, since it refers to the infinite totality of all possible interpretations. The question is nevertheless a natural one for Hilbert to ask, since it equates an infinitistic notion with a purely finitary, combinatorial notion.

In the 1930s, decisive progress was made on all three problems described above. As an unexpected bonus, there emerged for the first time, a completely precise and absolutely general notion of a mechanical or algorithmic procedure. In the following sections, we shall describe the dramatic developments of this decade.

3 Gödel's Theorems

In 1930 and 1931, Kurt Gödel solved the completeness problem for predicate logic, and made a basic contribution to the consistency problem. It was in the course of trying to solve the latter problem that he made the unexpected discovery that any axiomatic system containing a certain minimal part of number theory was necessarily incomplete.

Gödel's proof that the formal rules originally given by Frege are complete for the semantical concept of logical validity in first-order logic was published in 1930 (Gödel 1986: 102–23). Since then, many different versions of the proof have been given. Perhaps the most intuitively understandable form of the proof is to consider it as arising from the systematic search for a counter-model. This is the approach adopted when using the well-known formalism of semantic tableaux – currently employed in many introductory texts.

In the approach to completeness using analytic tableaux, the basic formalism consists of a tree labeled with sentences. We label the root of the tree with the negation of the formula in which we are interested. Each branch in the tree can then be considered as part of a search for a model that makes the negated formula labeling the root true. For example, if a branch in the tree contains an existential sentence $\exists x\, Fx$, then we extend the branch by adding an instance Fa. Similarly, if a branch contains a disjunction $A \vee B$, then we split the branch into two branches, one containing A, the other B. This search for a model must be carried out in a systematic way – we omit the details here. Provided the search is in fact systematic, then completeness can be seen to hold in the following sense. Either the search ends in failure, so that each branch terminates with an explicit contradiction, or this does not happen, in which case a model can be seen to exist. In the first case, the labeled tree is a proof of validity of the starting formula. For the details of this version of Gödel's completeness theorem, the reader is referred to Smullyan's elegant monograph of 1968.

The completeness theorem constitutes a vindication of Hilbert's formalist program, since it gives a purely syntactical, formal equivalent for the non-constructive concept of semantical validity. Gödel's next result, his great incompleteness theorem, threw in doubt most of Hilbert's formalist tenets.

In his original conception of the formalist program, Hilbert seems to have assumed implicitly the completeness of the axiomatic systems from which he began. The empirical evidence for this assumption was overwhelming. The axiomatic systems for number theory and analysis employed by the Hilbert school were more than adequate for formalizing all of the basic mathematics of the day, and more abstract topics such as functional analysis, the theory of transfinite cardinals and point set topology were all easily accommodated in the formal system of set theory created by Hilbert's colleague Ernst Zermelo. It was a shock, then, when Gödel showed that even elementary number theory is essentially incompletable.

Gödel's famous first incompleteness theorem can be stated as follows. Let T be a formal system of number theory so that all its theorems are true, and in which the predicate "s is a sequence of formulas constituting a proof of the formula A in the system T" is decidable, that is, there is an algorithm to decide for a given sequence s and formula A whether or not the relation holds. Then if T contains a certain minimum amount of

elementary number theory, it is incomplete, which is to say, there is a sentence G of T so that neither G nor its negation is a theorem.

Gödel's proof of the theorem (Gödel 1986: 144–95) involves the construction of a self-referential sentence akin to the Liar paradox. Gödel's basic insight was that by a system of encoding ("Gödel numbering"), formulas of the number-theoretical language could be considered as themselves being numbers (more precisely, Gödel's encoding produces an isomorphic image of the logical system in the natural numbers). In particular, we can express in the system T a number-theoretical relation $P(x, y)$ expressing the fact that x is the code number of a sequence of formulas that is a proof in the system T of the formula with code number y. Furthermore, since we have assumed that the proof predicate of T is decidable, the relation P is decidable in T, that is to say, if for particular numbers m, n, the relation $P(m, n)$ holds, then the sentence $P(\mathbf{m}, \mathbf{n})$ is provable in T, where '$\mathbf{m}$' is the numeral denoting the number m, and if it does not hold, then $\neg P(\mathbf{m}, \mathbf{n})$ is provable in T. Gödel can hence express the predicate 'x is the code number of a formula provable in T' as the existential formula $\mathrm{Prov}(x) \leftrightarrow \exists y P(y, x)$.

Gödel completes the proof of his first incompleteness theorem by making use of a clever diagonal construction to construct a self-referential sentence G that (interpreted via the coding devices) says of itself that it is not provable. More formally, we have as a theorem of T:

$$G \leftrightarrow \neg \mathrm{Prov}(\ulcorner G \urcorner),$$

where $\ulcorner G \urcorner$ stands for the code number of the sentence G itself. Gödel can now show that neither G nor its negation is a theorem of T by an argument resembling the reasoning leading to the contradiction in the Liar paradox. In this case, though, the paradoxical argument leads to incompleteness, not a contradiction.

We assumed in the above sketch of Gödel's argument that all of the theorems of T were true. However, an examination of the details of the proof shows that in demonstrating G itself to be unprovable, it is sufficient to assume that T is consistent. (A few years after Gödel's result appeared, Rosser using a more complicated self-referential sentence showed that the assumption of simple consistency was sufficient for incompleteness.) What is more, the argument showing this has a constructive, in fact finitary, character. It can therefore be formalized in T itself (since we assumed that T is adequate for elementary number theory). Thus the implication

$$\mathrm{Con}(T) \rightarrow \neg \mathrm{Prov}(\ulcorner G \urcorner)$$

is provable in T, where $\mathrm{Con}(T)$ is the statement formalizing the consistency of T. However, since the consequent of this implication is provably equivalent to G itself in T, it follows that if T itself is consistent, then $\mathrm{Con}(T)$ is unprovable in T.

This last result, known as Gödel's second incompleteness theorem, clearly has strong negative implications for Hilbert's consistency problem. If we start from a system of mathematics that encompasses the usual elementary forms of reasoning in number theory, then it presumably should include all of finitary reasoning (there is some uncertainty here, since Hilbert's notion of 'finitary reasoning' is not completely clear). But then if the system is consistent, it cannot prove its own consistency, and so a proof of

its consistency by finitary means is impossible. Hence it appears that Gödel's second incompleteness theorem precludes mathematics pulling itself up by its own bootstraps, as Hilbert had hoped.

Within the space of two years, Gödel had answered two of the fundamental problems of the Hilbert school. There remained the decision problem, though after the negative results of the incompleteness paper, it seemed most unlikely that this problem could have a positive solution. Gödel himself, in fact, came very close to providing a negative solution in the later sections of his incompleteness paper, a part that is rarely cited, since it was overshadowed by the later results of Church and Turing. However, these results are of considerable philosophical interest.

It was pointed out above that the completeness problem is not formulable in finitary terms, since it contains the non-constructive concept of semantic validity. However, it is possible to imagine constructive analogues of the completeness problem. To be more specific, let us imagine a formula of first-order predicate logic containing a certain number of predicate and relation symbols. We might then ask whether a constructive analogue of Gödel's completeness theorem holds in the sense that for any such formula, either it is provable by the usual axioms and rules for predicate logic, or, if it is not provable, we can find mathematical predicates (say, relations and predicates definable in number theory) so that when they are substituted for the atomic predicates in the formula, the resulting mathematical formula is refutable in some fixed axiom system for mathematics.

Gödel showed by analysing his unprovable formula G that for any consistent formal system S for mathematics, there are unprovable formulas of predicate logic that cannot be shown to be invalid by the substitution method in S. Looked at from the foundational point of view, this shows that the attempted constructive reformulation of the completeness problem fails. It also shows that it is highly unlikely that there could be an algorithm for the decision problem that could be proved to be correct in a standard system for mathematics.

The techniques that Gödel employed in the proof just described are essentially the same as those used a few years later by Church and Turing in showing the decision problem unsolvable. However, Gödel himself did not draw this conclusion. The difficulty lay in the fact that there was at that time no accepted precise definition delineating the class of mechanical procedures or algorithms. The creation of this definition was the next great step forward in logic, and is described in the next section.

4 Computability

Hilbert expected a positive solution to the decision problem, so that he was content to formulate the problem in terms of the intuitive mathematical notion of an algorithm. Gödel's incompleteness results, though, clearly pointed towards the conclusion that the problem was in fact unsolvable. To prove a negative result, however, it was essential to give a precise mathematical delineation of the concept of a mechanical procedure, or algorithm. This was first achieved by Alonzo Church and Alan Turing in 1936–37. Although Church was first in proposing a precise definition of computability (so that the identification of the intuitive with the mathematical concept is usually called

'Church's thesis'), Turing's conceptual analysis is usually held to be superior, and we shall follow Turing here. The reader is referred to an article by Wilfried Sieg for a penetrating account of the historical background to the work of Church and Turing (Sieg 1997).

Turing proceeded by giving a conceptual analysis of mechanical computation; the intended notion is that of a human carrying out the steps of an algorithm (recall that when Turing was writing, digital computers did not yet exist). His analysis can be explained in terms of two basic types of conditions, *boundedness conditions* and *locality conditions* (the terminology is that of Sieg). The computer (in the 1930s, when Turing was writing, this was always taken to denote a human being) operates in discrete time steps in a discrete symbol space – one can imagine a two-dimensional space, like a sheet of paper, or a one-dimensional space, like the paper tape of a Turing machine. The computer can perform the elementary actions of changing observed symbols and changing the set of observed symbols (moving in the symbol space). The boundedness conditions are these: the computer can recognize only finitely many distinct symbols, and has only a finite number of internal mental states (these are computational states, and need not be taken as mental states in a broader sense). The locality conditions are these: at each step, only finitely many symbols are observed, and in a single step, the computer can only move to a new symbol that is within a bounded distance of a previously observed symbol.

Turing adds to this model a deterministic condition: the elementary actions performed at each time step are uniquely determined by the current internal state, and the currently observed symbol configuration. To specify the functions computed by such a device, we need to add some conventions on input and output. With this, we have a complete model for mechanical computation.

Turing argued that a mechanical model for computation such as we have described in general terms above, is equivalent to the special case where the symbol space is one-dimensional, and at each step, exactly one symbol in this space is being observed. This is the well-known model of the *Turing machine*. The conceptual analysis sketched above is convincing evidence that this model is in fact a *universal* model for computation, in the sense that any mathematical function computed by an algorithm can be given in the form of a Turing machine.

Assuming this analysis of computation, we can now give a completely general definition of formal system, a concept that underlies Hilbert's conception of his program. A formal axiomatic system, then, is one in which there is a mechanical procedure to determine whether a string of symbols represents a meaningful assertion, and there is a set of axioms and rules that are also mechanically checkable (that is to say, there is an algorithm to determine whether or not a given string of symbols is or is not a proof in the system). With this definition, it is possible to state and prove a completely general version of Gödel's incompleteness theorem.

We can define the function $f_M(n)$ computed by a machine M as follows. We shall say that M *halts* if in the course of a computation it reaches a combination of internal state and input symbol for which it has no instruction. We shall suppose that the input and output of the machine consist of numerals in decimal notation. If M is given a number n as input, then if M eventually halts with the decimal notation for a number o written on the tape, then we say that o is the value of f_M for the input n. Notice that in general,

f_M is only a *partial* function, since there may be numerical inputs for which M goes into an infinite loop, for example, and never halts. We can define computable functions with two or more inputs in the same way.

It is now relatively easy to prove the undecidability of the decision problem. Every Turing machine can be specified by a finite list of instructions having a form such as: 'If you are in state 3, and are looking at the symbol 1, then change it to a 0, and go left one square.' These can be encoded as single numbers, using the technique of Gödel numbering, so that we can speak of the Turing machine M_k with code number k. We define the *halting problem* to be the problem of deciding, given two numbers k and n, whether the machine M_k eventually halts, when given input n.

We can prove the halting problem unsolvable by a straightforward diagonal argument. Let us suppose that the halting problem is solvable. Then there must be a Turing machine M that when given the pair of numbers k and n, outputs 1 if the machine M_k eventually halts, when given input n, otherwise 0. We can then use M to construct a new machine M' that when given the single input k, halts if M_k given k as input fails to halt, and otherwise fails to halt. (The details of the construction of M' from M are an exercise in Turing machine programming that we leave to the reader.) But now let h be the code number of the machine M', so that $M' = M_h$. Then on input h, M_h halts if and only if it does not halt, a contradiction.

Given the basic undecidability result we have just proved, we can now show the decision problem unsolvable. The proof is essentially a large scale exercise in formalizing assertions in first order logic. That is to say, given a machine M and input k, we can write down a formula $F(M, k)$ of first order logic that is valid if and only if M halts on input k. It follows that the decision problem must be unsolvable, since any algorithm solving it would lead to an algorithm solving the halting problem.

Our proof of unsolvability of the halting problem has another welcome corollary; another proof of Gödel's theorem. Let S be a standard formal system of number theory. We can formalize the encoding of Turing machines in S, so that we can, for example, write down a formula of S that expresses the fact that a number is a code number of a Turing machine. Using methods similar to those used in proving the decision problem unsolvable, we can find a formula $H(x, y)$ so that $\mathrm{H}(\mathbf{k}, \mathbf{n})$ is true if and only if the machine M_k halts on input n. Now we claim that S, if consistent, cannot prove all true statements of the form $H(\mathbf{k}, \mathbf{n})$ or $\neg H(\mathbf{k}, \mathbf{n})$. For suppose that it did; then we could solve the halting problem as follows. We can write a programme to print out one by one all the theorems of S (this is because we assumed that S is a formal system). Then to decide whether or not machine M_k halts on input n, we simply have to wait to see whether $H(\mathbf{k}, \mathbf{n})$ or $\neg H(\mathbf{k}, \mathbf{n})$ emerges as a theorem. This is impossible, so S is necessarily incomplete with respect to this class of statements. In fact, we can be a little more specific; there must be a particular machine M_k and input n so that M in fact does not halt on input n, but S cannot prove the statement $\neg H(\mathbf{k}, \mathbf{n})$.

A striking property of Turing's definition is that it is absolute, that is to say, it does not depend on the details of formalism used to define it. This aspect was stressed by Gödel in remarks at Princeton in 1946 commenting on an address by Tarski:

> Tarski has stressed in his lecture (and I think justly) the great importance of general recursiveness (or Turing's computability). It seems to me that this importance is largely due to

> the fact that with this concept one has for the first time succeeded in giving an absolute definition of an interesting epistemological notion, i.e., one not depending on the formalism chosen. (Gödel 1990: 150)

The evidence for Church's thesis is overwhelming, both in the sense that all known functions that are intuitively computable are computable in the sense of Turing, but also in the sense that many other proposed definitions (such as general recursiveness, computability by Markov algorithms, computability by register machines and so on) are equivalent to Turing's definition. One might object, of course, that (as was pointed out above) in the late 1930s the empirical evidence that the currently accepted formal systems for mathematics were complete was also overwhelming. In the case of Church's thesis, however, we have available Turing's conceptual analysis sketched above showing that any alternative concept of computability must drop the boundedness and locality conditions. A robust notion of quantum computation has recently emerged, that involves dropping (in a sense) the locality condition. This new notion does not lead to quantum computable functions that are not Turing computable, but it does open the door to possibly large gains in efficiency based on the exploitation of new features due to non-classical quantum effects.

5 Absolute and Relative in Logic

Hilbert's program was based on a belief that all mathematical concepts and constructions can be fully mirrored by formal, syntactical methods. Most of the results we have discussed above show that such mirroring is in fact impossible. For example, the concept of truth in number theory cannot be fully represented by provability in any formal system. In a similar way, we can show that many other mathematical concepts, such as the concept of a definable object, share the same essential incompletability with the notion of mathematical truth. In all of these cases, the incompleteness is a manifestation of the diagonal method. Any attempt to characterize the concept in a fixed formal framework leads by diagonalization to the construction of an object falling outside the formal characterization.

It may be asked whether one could not recover the absolute character of logical concepts by loosening the stringent finitistic character of Hilbert's requirements. A strategy of this sort was considered by Gödel in his 1948 Princeton lecture quoted above. In the case of definability, the argument of Richard's paradox of the least indefinable ordinal number, makes it clear that any absolute notion of definability must take all ordinal numbers as definable. Gödel's suggestion was to take definability in terms of ordinals as a possible definition of absolute definability. That is to say, a set is said to be ordinal definable if there is a sentence of the extended language of set theory in which all ordinals are primitive constants that uniquely defines it in the universe of all sets. This definition has the required property that it is impossible to apply the diagonal argument to find a set that is not definable; trivially, all ordinals are definable, so that the argument of Richard's paradox does not apply. On the other hand, the concept of definable object is obviously highly non-constructive, about as far from Hilbert's finitistic ideas as one can imagine.

It may seem surprising that an absolute concept, that of computability, emerged in the 1930s, a time when most of the concepts of logic, such as provability, were shown to have a relative, not absolute character. In fact, the absoluteness of this concept rests on the assumed absoluteness of another concept, namely the concept of truth for statements of number theory. This can be seen if we look at the definition of what it means for a Turing machine M to compute a function of natural numbers. This can be stated formally as: 'For every input n, there is a computation of M that terminates with a number o as output.' This can be encoded as a universal-existential sentence of elementary number theory. However, we cannot replace the notion of arithmetical truth here with a weaker notion such as the provability of the formalized version of the statement in an axiomatic system for number theory. By arguments similar to those used above in connection with the halting problem, we can show that no such system can prove all and only the true statements of this type. This is yet another manifestation of the incompleteness phenomenon.

Since we do seem to have a 'clear and distinct perception' of the notion of truth in number theory, it has often been argued that this demonstrates a clear superiority of humans over machines. More exactly, the incompleteness and undecidability results of Gödel, Church, and Turing have been held to show that humans have an absolute advantage over machines in that they are able to surpass any fixed machine in their insight into mathematical truths. The best known arguments for this conclusion are due to Lucas (1961) and Penrose (1989).

The Lucas/Penrose argument runs as follows. Let us suppose that we have programmed a computer to print out the theorems of a formal system of number theory one by one (the fact that we can program a computer to do this can be taken as an alternative definition of 'formal axiomatic system'). Gödel's incompleteness theorem applies to the formal system in question, so that there is for any such system a sentence G (the Gödel sentence for the system), that must be unprovable, provided the system is consistent. However, *we*, standing outside the formal system, and using our mathematical insight, can see that the sentence G is true, and so we can surpass the capacity of any fixed machine. This, according to Lucas and Penrose, proves that mechanical models of the mind are impossible, in short, that our minds cannot be machines.

The problem with the Lucas/Penrose argument presented above is that the key premise asserting that we can see the Gödel sentence to be true, remains undemonstrated. In fact, there are good reasons for thinking it to be false. The Gödel incompleteness theorem asserts a hypothetical proposition, namely that *if* the system in question is consistent, then the sentence G is unprovable. However, this hypothetical is provable in the system itself, under quite weak assumptions – in fact, this is the key idea of Gödel's second incompleteness theorem. For Lucas and Penrose to prove their case, they have to argue that we can see G itself to be true. This entails that we are able to show the system consistent.

There is no good reason to think that this last assumption is true. There are unsolved problems of mathematics (the Riemann hypothesis is perhaps the best known case) that have the property that if they are false, then this can be demonstrated by a simple counterexample. It follows from this that if we add such assumptions to a formal axiomatic system of mathematics, then the system is consistent if and only if the conjecture is true. This means that proving the consistency of a system based on, say, a version of

analytic number theory together with the Riemann hypothesis would be equivalent to proving the Riemann hypothesis. The Riemann hypothesis, though, is one of the most famous unsolved problems of mathematics, and it is unclear whether or not it will be solved in the near future. Lucas's and Penrose's assertion of an absolute superiority of minds over machines, then, seems to be without foundation.

Gödel himself tried to draw philosophical consequences from his incompleteness theorem, but was well aware that the simple argument of Lucas and Penrose was inadequate, since it rests on the unsupported assertion that human mathematicians can resolve all mathematical problems of a certain type. His most extended attempt at spelling out the philosophical implications of his theorem is to be found in his Gibbs lecture, delivered in 1951, but first published in the third volume of his collected works (Gödel 1995). Gödel's conclusion takes the form of a disjunction. If we make the assumption that humans can indeed resolve all consistency questions about formal systems of number theory, then an absolute superiority of humans over machines follows by the Lucas/Penrose argument. However, if this assumption is in fact false, then it follows that there must be mathematical assertions of a fairly simple type (since consistency assertions can be expressed, through the device of Gödel numbering as problems of number theory) that are absolutely unsolvable. In Gödel's own words:

> Either mathematics is incompletable in this sense, that its evident axioms can never be comprised in a finite rule, that is to say, the human mind (even within the realm of pure mathematics) infinitely surpasses the powers of any finite machine, or else there exist absolutely unsolvable diophantine problems of the type specified.
>
> (Gödel 1995: 310)

Gödel's own philosophical argument is not open to the simple objection made above to the Lucas/Penrose argument. However, one might still object that it involves an unjustified idealization of the human capacity for proving theorems. In particular, Gödel presupposes that humanly provable mathematical propositions form a well-defined set. However, one could argue that the totality of humanly provable propositions is a very ill-defined collection, with vague boundaries, quite unlike the set of theorems of a formal system.

The philosophical consequences of the incompleteness theorems in the broad sense remain obscure and controversial. In the narrower sense, though, Gödel's results provide a fairly conclusive refutation of Hilbert's formalist program in the foundations of mathematics. This is a rare and very unusual instance of decisive progress in the foundations of mathematics and logic.

References

Ewald, W. B. (ed.) (1996) *From Kant to Hilbert: A Source Book in the Foundations of Mathematics*, vol. 2. Oxford: Clarendon Press.

Gödel, K. (1986) *Collected Works*, vol. 1: *Publications 1929–1936*, Solomon Feferman, John W. Dawson, Jr., Stephen C. Kleene, Gregory H. Moore, Robert M. Solovay, and Jean van Heijenoort (eds.). Oxford: Oxford University Press.

Gödel, K. (1990) *Collected Works*, vol. 2: *Publications 1938–1974*, Solomon Feferman, John W. Dawson, Jr., Stephen C. Kleene, Gregory H. Moore, Robert M. Solovay, and Jean van Heijenoort (eds.). Oxford: Oxford University Press.
Gödel, K. (1995) *Collected Works*, vol. 3: *Unpublished Essays and Lectures*, Solomon Feferman, John W. Dawson, Jr., Warren Goldfarb, Charles Parsons and Robert M. Solovay (eds.). Oxford: Oxford University Press.
Lucas, J. R. (1961) Minds, machines and Gödel. *Philosophy*, 36, 112–27.
Penrose, R. (1989) *The Emperor's New Mind*. Oxford: Oxford University Press.
Sieg, W. (1997) Step by recursive step: Church's analysis of effective calculability. *Bulletin of Symbolic Logic*, 3, 154–80.
Smullyan, R. (1968) *First-Order Logic*. Berlin: Springer (reprinted by Dover Publications 1995).
Van Heijenoort, J. (1967) *From Frege to Gödel*. Cambridge, MA: Harvard University Press.

Further Reading

Boolos, G. and Jeffrey, R. C. (1989) *Computability and Logic*, 3rd edn. Cambridge: Cambridge University Press.
Kleene, S. C. (1952) *Introduction to Metamathematics*. New York: Van Nostrand.
Reid, C. (1970) *Hilbert*. Berlin: Springer-Verlag.